WEST

Help | Logout | Interrupt

Main Menu | Search Form | Posting Counts | Show S Numbers | Edit S Numbers | Preferences | Cases

Search Results -

Terms	Documents
L73 and (relocat$ near data or move near data)	88

Database:
US Patents Full-Text Database
US Pre-Grant Publication Full-Text Database
JPO Abstracts Database
EPO Abstracts Database
Derwent World Patents Index
IBM Technical Disclosure Bulletins

Search: Refine Search

Recall Text | Clear

Search History

DATE: Tuesday, January 21, 2003 Printable Copy Create Case

Set Name side by side	**Query**	**Hit Count**	**Set Name** result set
	DB=USPT,PGPB,JPAB,EPAB,DWPI,TDBD; PLUR=YES; OP=OR		
L74	L73 and (relocat$ near data or move near data)	88	L74
L73	account$ same report	5184	L73
L72	L68 and report same data same relocat$	4	L72
L71	L68 and data near relocat$	1	L71
L70	L68 and relocat$ same data	9	L70
L69	L68 and relocat$ near data	1	L69
L68	account$ near (payable or AR or AP)	621	L68
L67	L66 and (account$ same report or account$ near AP or AR)	14	L67
L66	(report same segmentation or data near id near location)	189	L66
L65	L63 and (accounting or AP or AR)	23	L65
L64	l63 and accounting same report	3	L64
L63	report same segmentation	158	L63
L62	L60 and mapping and translation	12	L62
L61	L60 and mapping and translation adj processes	0	L61
L60	monarch	2444	L60

L59	L58 and mapp$ and translation same process	19	L59
L58	mercator	346	L58
L57	data adj junction	125	L57
L56	l55 and account$ near (payable or AR or AP)	2	L56
L55	(data adj junction or mercator or monarch)	2913	L55
L54	6397191.uref.	0	L54
L53	l35 and 5325290.pn.	0	L53
L52	l35 and L51	0	L52
L51	5325290.pn.	2	L51
DB=USPT; PLUR=YES; OP=OR			
L50	5383113.pn.	1	L50
L49	4667292.pn.	1	L49
L48	5325290.pn.	1	L48
L47	5359509.pn.	1	L47
L46	5361202.pn.	1	L46
L45	5383113.pn.	1	L45
DB=USPT,PGPB,JPAB,EPAB,DWPI,TDBD; PLUR=YES; OP=OR			
L44	l35 and account$ near (payable or AR or AP)	2	L44
L43	L35 and l42	0	L43
L42	workflow and account$ near (payable or AR or AP)	94	L42
L41	L40 and account$ near (payable or AR or AP)	0	L41
L40	verve or embedded same workflow same system	63	L40
L39	verve same embedded same workflow same system	0	L39
L38	L37 and verve same embedded same workflow same system	0	L38
L37	(sqr near server or sqrib or webi)	14	L37
L36	L35 and (sqr near server or sqrib or webi)	0	L36
L35	(data near junction or mercator or monarch)	3047	L35
L34	l28 and report	223	L34
L33	L28 and workflow same report	5	L33
L32	l26 and L28	0	L32
L31	l25 and L28	0	L31
L30	l8 and L29	6	L30
L29	l1 and L28	184	L29
L28	L27 and account$ near (payable or AR or AP)	315	L28
L27	account$ near receivable	776	L27
L26	workflow near report	31	L26
L25	(((707/507)!.CCLS.))	31	L25
L24	(((707/506)!.CCLS.))	3	L24
L23	(((707/505)!.CCLS.))	2	L23
L22	(((709/219)!.CCLS.))	1609	L22
L21	(((709/203)!.CCLS.))	2747	L21
L20	(((709/$)!.CCLS.))	22269	L20

L19	(((340/542)!.CCLS.))	458	L19
L18	(((340/426)!.CCLS.))	1146	L18
L17	(((426/$)!.CCLS.))	81908	L17
L16	(((340/$)!.CCLS.))	100573	L16
L15	(((235/382)!.CCLS.))	847	L15
L14	(((235/375)!.CCLS.))	3655	L14
L13	(((235/$)!.CCLS.))	58480	L13
L12	(((707/501.1)!.CCLS.))	6	L12
L11	(((707/500)!.CCLS.))	5	L11
L10	(((707/104.1)!.CCLS.))	2090	L10
L9	(((707/1)!.CCLS.))	2104	L9
L8	(((707/$)!.CCLS.))	14012	L8
L7	(((705/40)!.CCLS.))	489	L7
L6	(((705/38)!.CCLS.))	399	L6
L5	(((705/36)!.CCLS.))	723	L5
L4	(((705/35)!.CCLS.))	763	L4
L3	(((705/7)!.CCLS.))	761	L3
L2	(((705/4)!.CCLS.))	378	L2
L1	((705/$)!.CCLS.)	17797	L1

END OF SEARCH HISTORY

WEST

□ Generate Collection | Print

L40: Entry 22 of 63 File: USPT May 28, 2002

US-PAT-NO: 6397191
DOCUMENT-IDENTIFIER: US 6397191 B1

TITLE: Object-oriented workflow for multi-enterprise collaboration

DATE-ISSUED: May 28, 2002

INVENTOR-INFORMATION:

NAME	CITY	STATE	ZIP CODE	COUNTRY
Notani; Ranjit N.	Irving	TX		
Parasnis; Abhay V.	Irving	TX		
Whipple; Mark B.	Dallas	TX		

ASSIGNEE-INFORMATION:

NAME	CITY	STATE	ZIP CODE	COUNTRY	TYPE CODE
i2 Technologies US, Inc.	Dallas	TX			02

APPL-NO: 09/ 154661 [PALM]
DATE FILED: September 18, 1998

PARENT-CASE:
RELATED APPLICATIONS This application is a continuation-in-part of U.S. patent Ser. No. 09/092,348, filed Jun. 5, 1998, now U.S. Pat. No. 6,119,149. This application is related to U.S. patent application Ser. No. 09,156,722 entitled "Computer Workspace Providing Event Management Based on a Permissibility Framework;" U.S. patent application Ser. No. 09/156,265 entitled "System and Method for Remotely Accessing Data;" U.S. patent application Ser. No. 09/156,264 entitled "Workflow Communication;" U.S. patent application Ser. No. 09/156,333 entitled "Workflow Synchronization;" U.S. patent application Ser. No. 09/156,334 entitled "Method and System for Managing Collaboration Within and Between Enterprises;" U.S. patent application Ser. No. 09/156,342 entitled "System and Method for Event Notification Through a Firewall;" and U.S. patent application Ser. No. 09/156,434 entitled "Exemplar Workflow Used in the Design and Deployment of a Workflow for Multi-Enterprise Collaboration;" all filed Sep. 18, 1998, all of which are currently pending, the disclosures of which are incorporated by reference herein.

INT-CL: [07] G06 F 17/60

US-CL-ISSUED: 705/9
US-CL-CURRENT: 705/9

FIELD-OF-SEARCH: 705/1, 705/7, 705/8, 705/9, 709/205, 700/99, 707/103

PRIOR-ART-DISCLOSED:

U.S. PATENT DOCUMENTS

Search Selected | Search ALL

	PAT-NO	ISSUE-DATE	PATENTEE-NAME	US-CL
□	5634127	May 1997	Cloud et al.	709/313
□	5745687	April 1998	Randell	709/201
□	5931900	August 1999	Notani et al.	709/200
□	6052684	April 2000	Du	707/8
□	6073109	June 2000	Flores et al.	705/8
□	6157941	December 2000	Verkler et al.	709/202

FOREIGN PATENT DOCUMENTS

FOREIGN-PAT-NO	PUBN-DATE	COUNTRY	US-CL
0 778 535	June 1997	EP	
WO 96/17296	June 1996	WO	
WO 99/10825	March 1999	WO	
WO 99/63469	December 1999	WO	

OTHER PUBLICATIONS

International Search Report, Jun. 30, 2000.
Mihai Barbuceanu and Mark S. Fox, Enterprise Integration Laboratory, University of Toronto: "Coordinating Multiple Agents in the Supply Chain", 1996, pp. 1335-141, IEEE, XP-002090170.
Sun Microsystems; "Innovative Java technology that simplifies distributed application development"; Internet; http://java.sun.com; downloaded Jan. 1999; all.
David Stirrup, et al.; "Workflow Management Coalition Advances Workflow Interoperability and Business Process Definition Interfaces"; Internet; http://www.aiim.org/wfmc; Feb. 26, 1996; all.
David Stirrup; "Workflow Management Coalition defines Interface between Workflow Engines and Process Definition Information"; Internet; http://www.aiim.org/wfmc; downloaded Jan. 1999; all.
David Hollingsworth; "Workflow Management Coalition The Workflow Reference Model"; The Workflow Management Coalition; Internet; http://www.aiim.org/wfmc; Jan. 19, 1995; all.
Sun Microsystems, Inc.; "Java.TM.Object Serialization Specification"; Sun Microsystems, Inc.; Internet; http://www.java.sun.com; Nov. 30, 1998; all.

ART-UNIT: 2161

PRIMARY-EXAMINER: Trammell; James P.

ASSISTANT-EXAMINER: Dixon; Thomas A.

ABSTRACT:

A computer implemented process is disclosed for enterprise collaboration. The process includes instantiating an object-oriented workflow where the object-oriented workflow comprises objects associated with activities to be performed within the workflow. The objects of the object-oriented workflow are then deployed across enterprise boundaries to nodes on which associated activities are to be performed. After deployment, the deployed objects are executed to provide multi-enterprise collaboration with the activities communicating data using objects that carry both data and behavior. Further, in one embodiment, the process further includes creating in-memory object models at nodes which objects executing at the nodes can access.

40 Claims, 22 Drawing figures

WEST

End of Result Set

□ Generate Collection | Print

L44: Entry 2 of 2 File: USPT Jun 6, 2000

DOCUMENT-IDENTIFIER: US 6073104 A
TITLE: System for invoice record management and asset-backed commercial paper program management

Detailed Description Text (13):
As shown in FIG. 5, which illustrates the first exemplary embodiment of system 30, the hardware configuration at each healthcare provider's location 40 includes a personal computer or workstation acting as a remote sentinel system 42. The currently preferred system 42 uses an Intel processor system, the "Microsoft Windows for Workgroups" operating environment, a database program comprising a runtime version of Microsoft's "Access" database program, Tools & Technologies' Data Junction program for capturing and translating data between various computer platforms, and International Software's Remote Office.TM. computer communication program. Sentinel system 42 preferably has two one-gigabyte hard drives and an internal fax/modem. The second one-gigabyte hard drive mirrors the first drive and serves as a backup. The sentinel system 42 is connected to the provider's local area network 44, and can download claim records and other information directly from the provider's mainframe or other computing facility 46, which executes the accounting system 36. The sentinel system 42 sends summary data to the central location 50.

Detailed Description Text (76):
FIG. 45 shows how the accounts payable, namely advance payable, interest payable and fees payable, are updated for each daily pool balance sheet by software 53. For each daily pool, the initial balance in the advance payable account is set equal to the amount advanced. For each day that the pool has a balance greater than zero in its advance payable account, an interest expense and a fee expense are calculated. On those days that the pool does not have any collections, the interest expense is added to interest payable on the balance sheet to create the new interest payable amount and interest paid on the cash flow statement is set equal to zero. Also, the fee expense is added to fees payable on the balance sheet to create the new fees payable amount and the fee paid on the cash flow statement is set equal to zero. If the pool has any collections, the collections are used to pay down accounts in the following order: interest, fees, and then advances. If the amount collected is less than today's interest expense plus the interest payable on the balance sheet, then the total amount collected is applied to interest and the new interest payable amount becomes the difference between today's interest expense plus the interest payable on the balance sheet and the amount collected. Interest paid on the cash flow statement equals the amount collected and fee paid is set equal to zero. If the amount collected is greater than or equal to that day's interest expense plus the balance in the interest payable account, then an amount equal to the interest expense plus the interest payable is used to pay off these accounts. Interest paid on the cash flow statement equals the day's interest expense plus the balance in the interest payable account. If the excess is less than today's fee expense plus the balance in the fee payable account, then the new fee payable amount equals today's fee expense plus the balance in the fee payable account minus the excess. Fee paid in the cash flow statement equals the excess. If the excess is greater than or equal to today's fee expense plus the balance in the fee payable account, then an amount equal to today's fee expense plus the balance in the fee payable account is used to pay off these accounts. Fee paid in the cash flow statement equals today's fee expense plus the balance in the fee payable account. If there is any remaining cash, the cash is used to pay down the advance payable account to the point where either all of the cash has been used or all of the advance payable has been paid off. The advance paid in the cash flow statement equals the reduction in the advance payable.

WEST

□ Generate Collection Print

L63: Entry 71 of 158 File: USPT Oct 19, 1999

US-PAT-NO: 5970464
DOCUMENT-IDENTIFIER: US 5970464 A

TITLE: Data mining based underwriting profitability analysis

DATE-ISSUED: October 19, 1999

INVENTOR-INFORMATION:

NAME	CITY	STATE	ZIP CODE	COUNTRY
Apte; Chidanand Vinayak	Chappaqua	NY		
Grossman; Edna	Chappaqua	NY		
Pednault; Edwin Peter Dawson	Cortlandt Manor	NY		
Rosen; Barry Kumin	Stormville	NY		
Tipu; Fateh Ali	Wappingers Falls	NY		
Wang; Hsueh-ju	Mohegan Lake	NY		
White; Brian Frederick	Yorktown Heights	NY		

ASSIGNEE-INFORMATION:

NAME	CITY	STATE	ZIP CODE	COUNTRY	TYPE CODE
International Business Machines Corporation	Armonk	NY			02

APPL-NO: 08/ 926804 [PALM]
DATE FILED: September 10, 1997

INT-CL: [06] G06 F 17/60

US-CL-ISSUED: 705/4; 705/7
US-CL-CURRENT: 705/4; 705/7

FIELD-OF-SEARCH: 705/4, 705/7, 705/10, 705/35, 705/36, 705/38, 705/2, 706/934, 706/12, 706/21, 706/60, 706/47, 707/104

PRIOR-ART-DISCLOSED:

U.S. PATENT DOCUMENTS

Search Selected Search ALL

	PAT-NO	ISSUE-DATE	PATENTEE-NAME	US-CL
□	4766539	August 1988	Fox	705/4
□	4975840	December 1990	DeTore et al.	705/4
□	5586313	December 1996	Schnittker et al.	707/104
□	5692107	November 1997	Simoudis et al.	706/12
□	5712984	January 1998	Hammond et al.	705/4
□	5752237	May 1998	Cherny	705/4
□	5781911	July 1998	Young et al.	707/201
□	5809478	September 1998	Greco et al.	705/4

OTHER PUBLICATIONS

Gallagher, Cecily, "Risk Classification Aided by New Software Tool,"National Underwriter Property & Casualty Risk Benefits and Management, No. 17, p.19, Apr. 27, 1992.
Derrig, Richard A., "Fuzzy Techniques of Pattern Recognition in Risk and Claims Classification," The Journal of Risk and Insurance, vol. 62 (Sept. 1995), pp. 447-482.
Bentley, Trevor, "Mining for Information," Management Accounting-London, vol. 75, No. 6 (Jun. 1997), p.56.

ART-UNIT: 271

PRIMARY-EXAMINER: Tkacs; Stephen R.

ASSISTANT-EXAMINER: Morgan; George D.

ABSTRACT:

A computer implemented method of underwriting profitability analysis delivers the analytic process to a wide cross section of insurance decision makers. The underwriting profitability analysis system leverages an existing investment in databases and improves underwriting business processes. Data mining techniques are applied to historical policy and claims to extract rules that describe policy holders with homogeneous claim frequency and severity characteristics. These rule sets are used to classify policy holders into distinct risk groups, each with its own set of characteristics, including pure premium. Breaking up a book of business into segments allows identification of sub-populations of policy holders that distinctly deviate from the expected normal pure premium. This identification allow the insurance business analysts to interactively adjust eligibility criteria and examine altered characteristics of the covered segments until satisfactory. The system is implemented on a client server using network centric language technology.

7 Claims, 14 Drawing figures

WEST

□ Generate Collection Print

L63: Entry 71 of 158 File: USPT Oct 19, 1999

DOCUMENT-IDENTIFIER: US 5970464 A
TITLE: Data mining based underwriting profitability analysis

Detailed Description Text (42):
The scenario analysis subsystem is a crucial piece in the UPA solution that ties in the data mining process to the business problems at hand. The idea here is to allow the user to determine the value of a P&C insurance product by specifying it to the system, and having the system provide critical business information about the product, segment by segment. The scenario analysis screen shown in FIG. 13 is displayed when the user selects the Scenarios tab. Using the scenario analysis screen, the user will specify a database to analyze, a data mining model to be used as the base, and a product/population identifier. A product identifier is essentially a user created rule set, that could either represent an existing product, or a new product under consideration. Once specified, and the "Analyze" button selected from the "File" pull down menu, the system will perform a scenario analysis, and display to the user, in a subsequent screen, a detailed segmentation report.

Detailed Description Text (44):
The scenario analysis result will first report on the gross statistics on how the product rule set covered the database, and within this coverage, using the base model, will be a detailed segmentation report that breaks down the coverage into individual segments, listed by the segments' coverage, percentage coverage, severity estimate, frequency estimate, pure premium, loss ration, and other entries that may be of interest. In addition, the screen will permit the table to be sorted by any of these columns. This "what-if" style scenario analysis will assist the users to identify problems and opportunities with existing as well as new P&C products.

WEST

□ Generate Collection Print

L74: Entry 71 of 88 File: USPT Mar 4, 1997

DOCUMENT-IDENTIFIER: US 5608874 A
TITLE: System and method for automatic data file format translation and transmission having advanced features

Detailed Description Text (202):
Payroll data has been automatically entered into the accounting software application general ledger, eliminating the need for this data to be entered manually. Subsequently, the subscriber enters the general ledger module of the accounting application software, and prints out an exception report. In this example, the exception report flags the subscriber that a certain employee received and inappropriate amount of money in his last paycheck, and that another certain employee's salary expense was allocated to a non-existent account number.

Detailed Description Text (203):
Bank statement data has been automatically entered into the bank book module of the accounting software application, eliminating the need for this data to be entered manually. The subscriber then enters the bank book module of the accounting software application and prints out reports to examine if any unusual transactions occurred. In this example scenario, a check that does not belong to the subscriber, and a check amount that does not agree with the bank statement amount have been flagged for the subscriber to review.

Detailed Description Text (212):
Data manipulations in the current invention may be metaphorically described as "manufacturing assembly line" style. The spirit of this feature of the invention is to move "raw data" into place, convert the "raw data" into usable data via processing stages, and finally move the "finished data" out for subscribers to use. Such an approach is useful because of the need to handle various pieces of raw and partially processed data with different programs created by, or associated with the present invention. The process includes an "assembly line" and "robots". The "assembly line" are the devices (both hardware and software) that move the data from one processing stage to the next processing stage. The "robots" are the programs comprising the current invention.

WEST

□ Generate Collection Print

L3: Entry 17 of 28 File: USPT Sep 17, 2002

DOCUMENT-IDENTIFIER: US 6453356 B1
TITLE: Data exchange system and method

Detailed Description Text (18):
Adapter A is configured to disassociate the informational content transported within the EDI data stream D.sub.1 from its associated EDI format and dialect. The EDI informational content extracted by adapter A is reformatted to a common representation and then transported through the data exchange engine 62 to a destination application within the inter-exchange carrier #2 environment. The adapter 120, in this embodiment, is configured to translate the EDI information content having a common format to an EDI format and dialect required by the destination application. Adapter 120 also converts source EDI information transmitted from inter-exchange carrier #2 into the common or generic form.

Detailed Description Text (20):
In FIG. 4, there is illustrated another embodiment of an information interchange environment within which the data exchange infrastructure in accordance with the present invention may find particular application. The data exchange infrastructure may be implemented to enhance workflow or process management systems which interact with any number of legacy or proprietary applications, remote data stores, or various user and application work queues. In this embodiment, the data exchange infrastructure provides for reliable application integration, data movement, and remote work queues. In this configuration, unreliable system implementations, such as screen scraping applications or networks with poor line condition, may be transformed into reliable implementations through use of the data exchange infrastructure. In particular, this unreliable to reliable conversion is achieved, for example, through the use of persistent queues, rollback processing upon transaction failures, which provides for transactional integrity, and transaction retry processing as necessary.

Detailed Description Text (108):
```
EXAMPLE #33 Assumes a populated DX_ListObject identified by a pointer "PlistObj" //
Instantiate the DX_ListObjectIterator for DX_ListObject pointer PlistObj
DX_ListObjectIterator PlistIter(*PlistObj); // Set the iterator to the beginning of
the list PlistIter.toFirst( ); DX_CommonBase *PcurrentObj=0;
PcurrentObj=(DX_CommonBase*)PlistIter.getCurrent( ); while(PcurrentObj) { // Do
something with object/attribute . . . PcurrentObj=(DX_CommonBase*) PlistIter.next(
); }
```

Detailed Description Text (197):
As was discussed previously, two types of priority based queues are used, namely, the incoming Receive Queues and the outgoing Send Queues. Each outgoing adapter will have its own outgoing queue so that any interface specific translation or routing may be performed outside the engine core. Each instance of the DX_Engine executable has one or more input queues, although only one is allowed for file-based queues, and one or more output queues. An instance of the DX_QueueManager class is used as a central proxy to all queue access, and will be mutex protected and record-lock protected, for file-based implementation, or row lock protected, for database implementations, to prevent data contention.

Detailed Description Text (215):
DX_FileSubQueue is mapped to a set of files of one queue of one priority. The internal DX_QueueObjectList object acts as a buffer at running time. All the

operations will not be written into files until commit or rollback time.

Detailed Description Text (223):
DX_DBQueue is the database-based implementation of the DX_Queue interface. Its instance is mapped to a single table per queue at run-time. The order of the records is determined by the enqueue time stamp. All dequeue operations are sorted by priority and enqueue time stamp. An illustrative example of a DX_DBQueue implementation is provided as follows:

Detailed Description Text (227):
The invocation of the Enqueue( ) and Dequeue( ) API is effected by sending a request to the Queue Manager Object. In response to a request, the Queue Manager Object locates the correct queue and populates the operations to that queue object. When Enqueue( ) is invoked, the object is placed into a buffer list and will not be collapsed into a data stream until Commit( ) is invoked. Until the commit time, the DX_IndexObject attribute of the DX_QueueOperation object retains the real meaning, which may be used in connection with a rollback operation if the commit operation is not successfully executed. For purposes of serialization, if the queue is a database, the row-level locking provided by the database is used. If the queue is a file, file access control is used. When a Common Object is serialized at commit time, a priority tag is appended to its private attribute list, such that when the Common Object is dequeued, its priority can be easily determined.